
# Performance Analysis with Vampir

**11th VI-HPS Tuning Workshop (MdS, Saclay)**
**22-25 April 2013**

Frank Winkler, Ronny Tschüter, Andreas Knüpfer
ZIH, Technische Universität Dresden

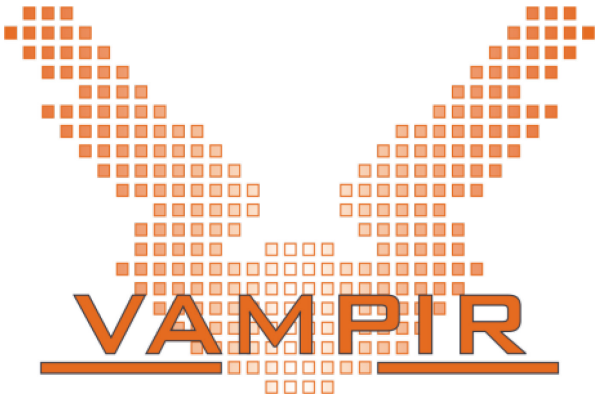
## Part I: Welcome to the Vampir Tool Suite

- Mission
- Event Trace Visualization
- Vampir & VampirServer
- The Vampir Displays

## Part II: Vampir Hands On

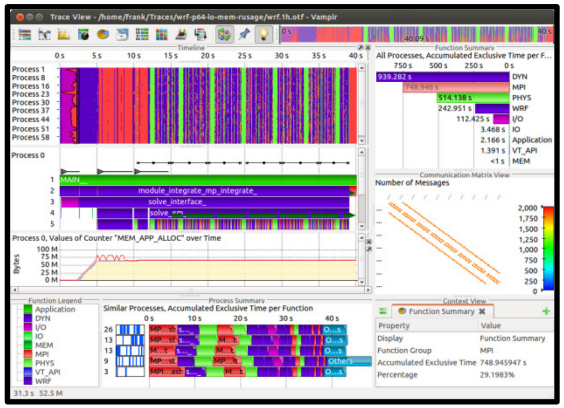
- Visualizing and analyzing NPB-MZ-MPI / BT

## Part III: Summary and Conclusion



# Mission

- Visualization of dynamics of complex parallel processes
- Requires two components
  - Monitor/Collector (Score-P)
  - Charts/Browser (Vampir)



## Typical questions that Vampir helps to answer:

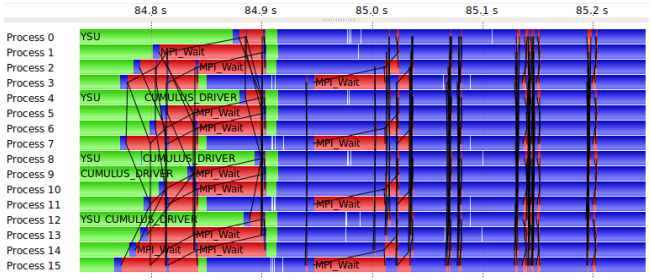
- What happens in my application execution during a given time in a given process or thread?
- How do the communication patterns of my application execute on a real system?
- Are there any imbalances in computation, I/O or memory usage and how do they affect the parallel execution of my application?

# Event Trace Visualization with Vampir

- Alternative and supplement to automatic analysis
- Show dynamic run-time behavior graphically at any level of detail
- Provide statistics and performance metrics

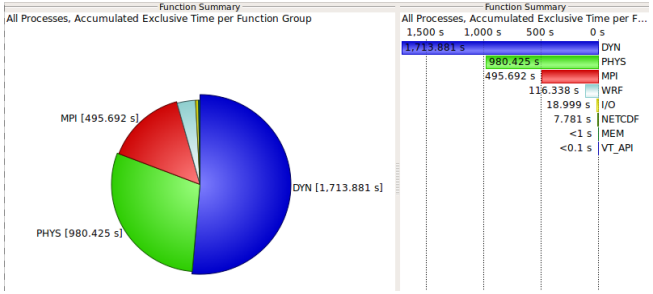
## Timeline charts

– Show application activities and communication along a time axis




## Summary charts

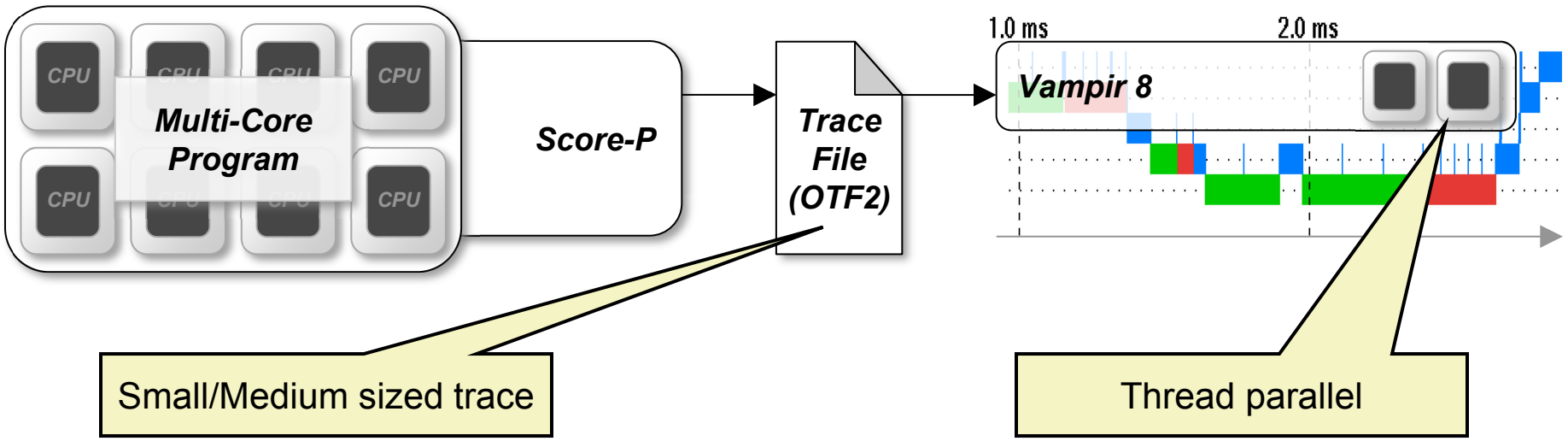
– Provide quantitative results for the currently selected time interval



# Vampir – Visualization Modes (1)

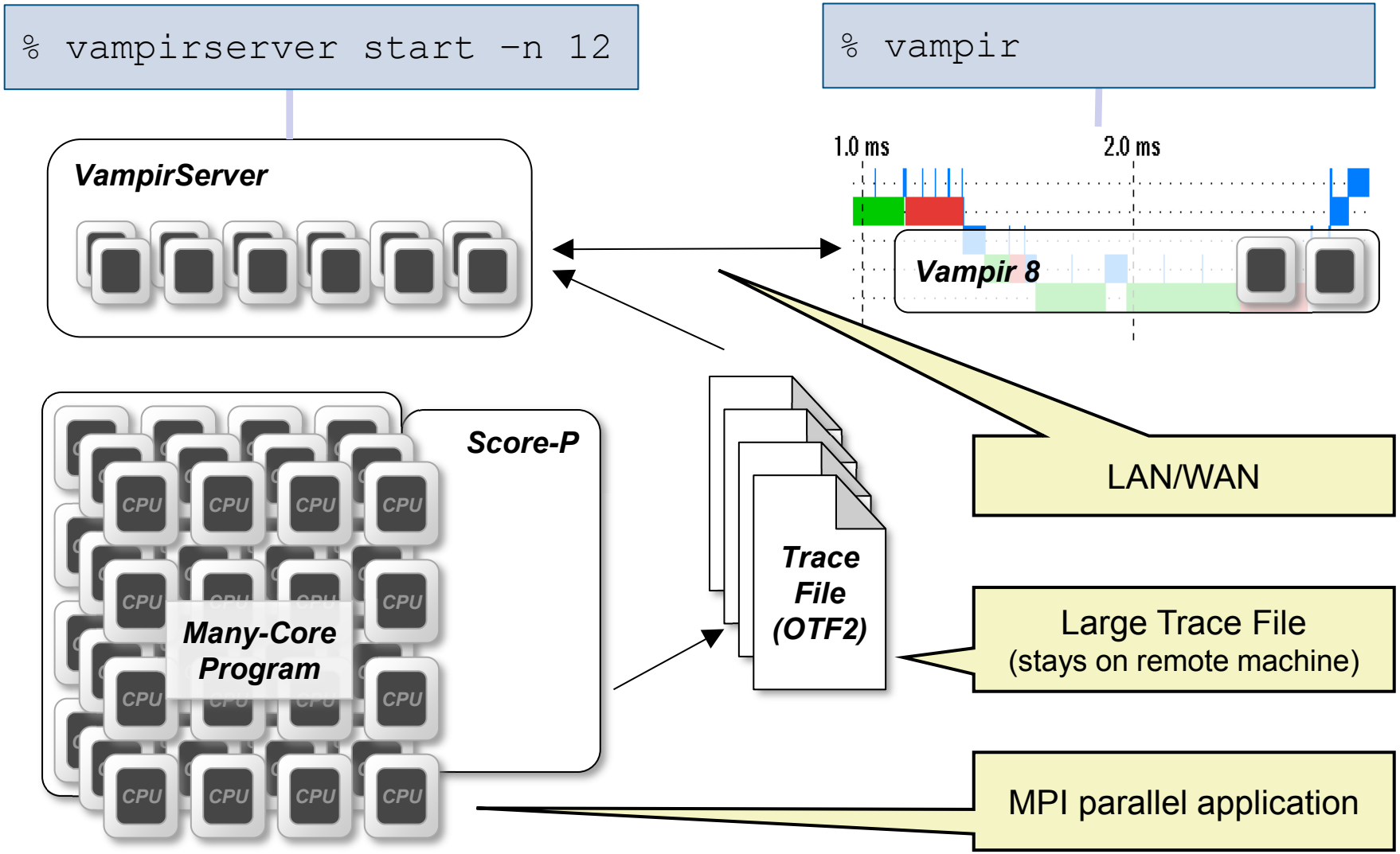
- Directly on front end or local machine

```
% vampir
```



# Vampir – Visualization Modes (2)

- On local machine with remote VampirServer



# Usage order of the Vampir Performance Analysis Toolset


1. Instrument your application with Score-P

2. Run your application with an appropriate test set

3. Analyze your trace file with Vampir
   - Small trace files can be analyzed on your local workstation
     1. Start your local Vampir
     2. Load trace file from your local disk
   - Large trace files should be stored on the HPC file system
     1. Start VampirServer on your HPC system
     2. Start your local Vampir
     3. Connect local Vampir with the VampirServer on the HPC system
     4. Load trace file from the HPC file system

# The main displays of Vampir

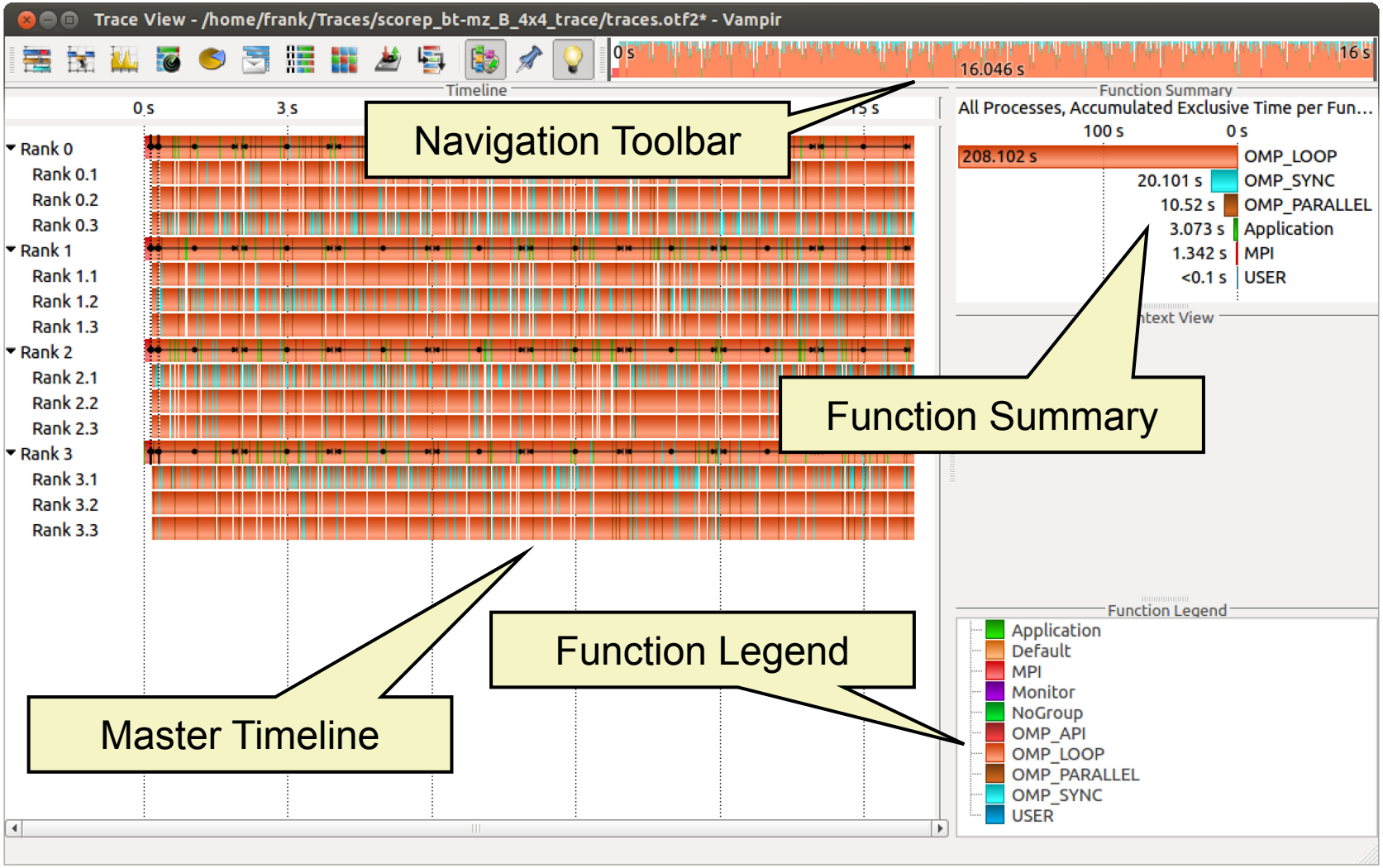
- **Timeline Charts:**
  -  Master Timeline
  -  Process Timeline
  -  Counter Data Timeline
  -  Performance Radar
- **Summary Charts:**
  -  Function Summary
  -  Message Summary
  -  Process Summary
  -  Communication Matrix View


# Vampir Hands on

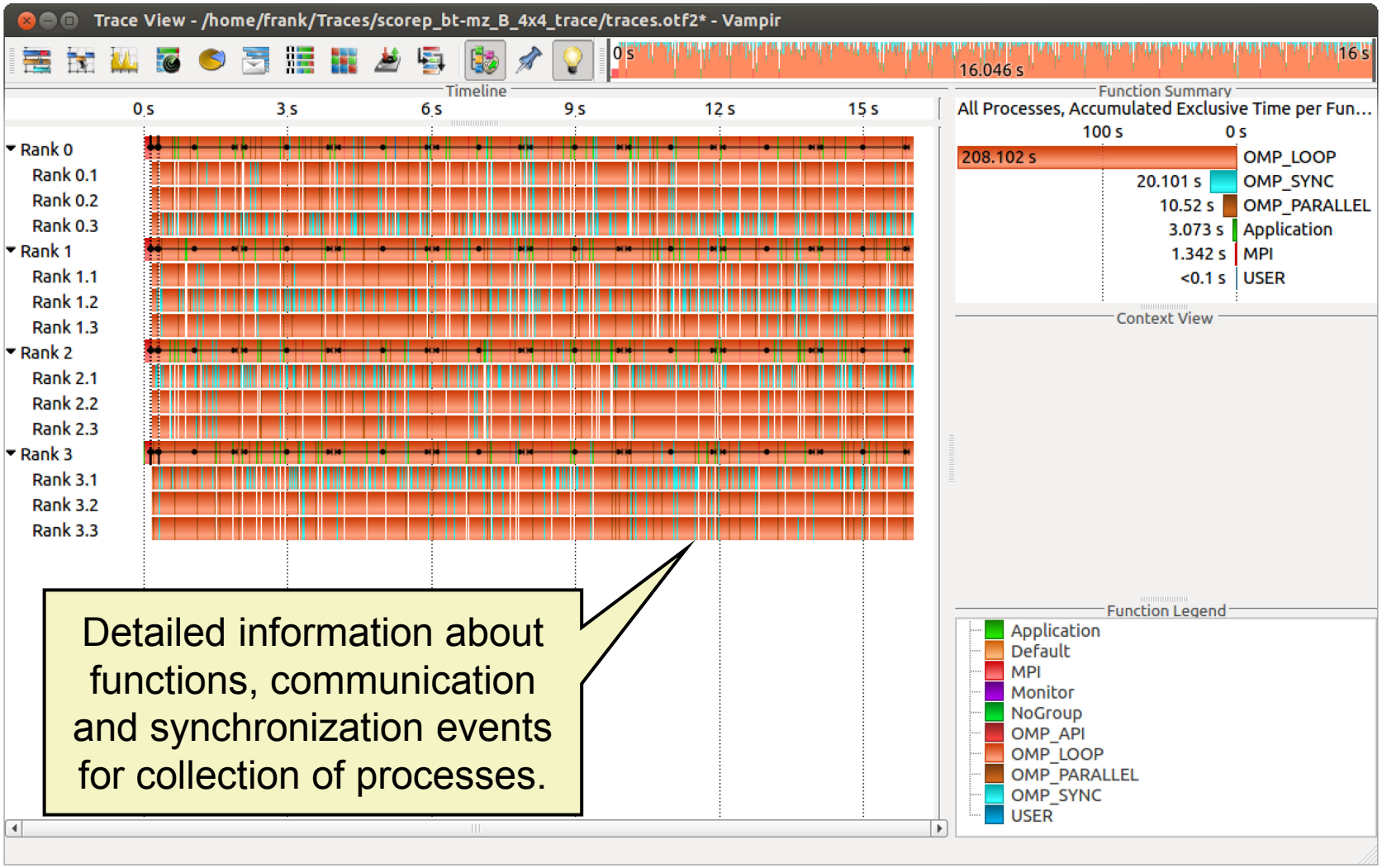
Visualizing and analyzing NPB-MZ-MPI / BT

# Vampir: Visualization of the NPB-MZ-MPI / BT trace

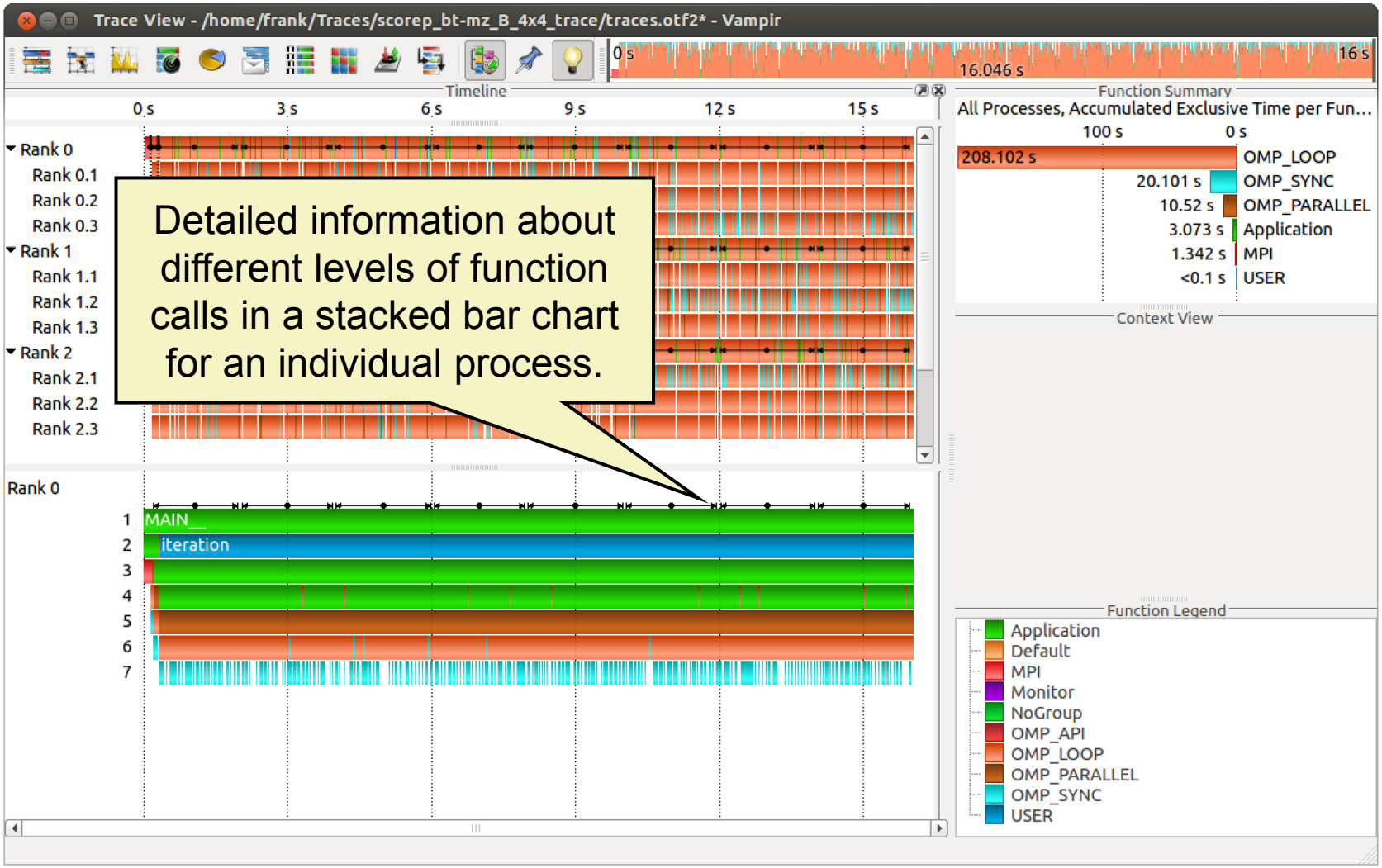
```
% vampir scorep_bt-mz_B_4x4_trace
```



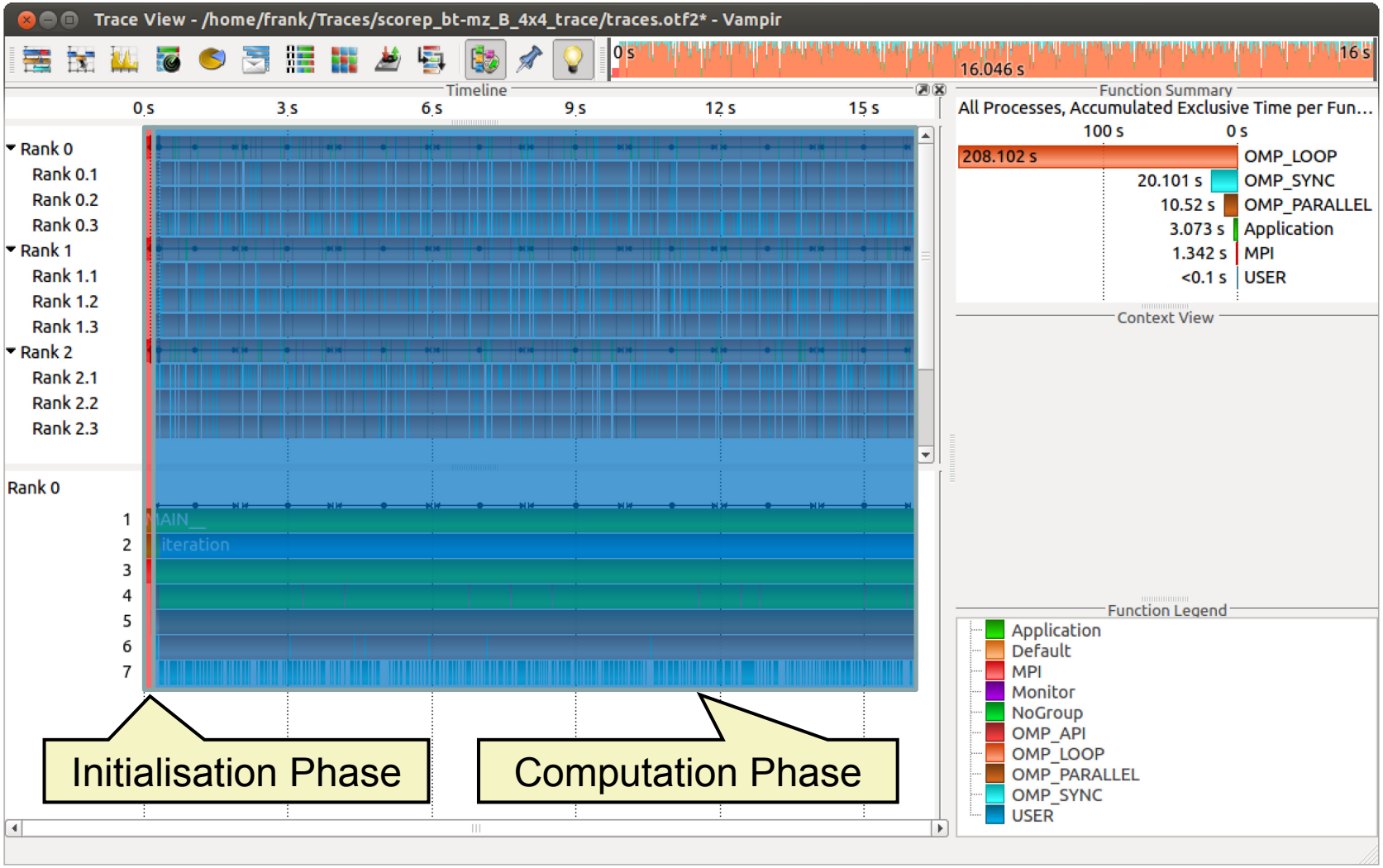
# Master Timeline



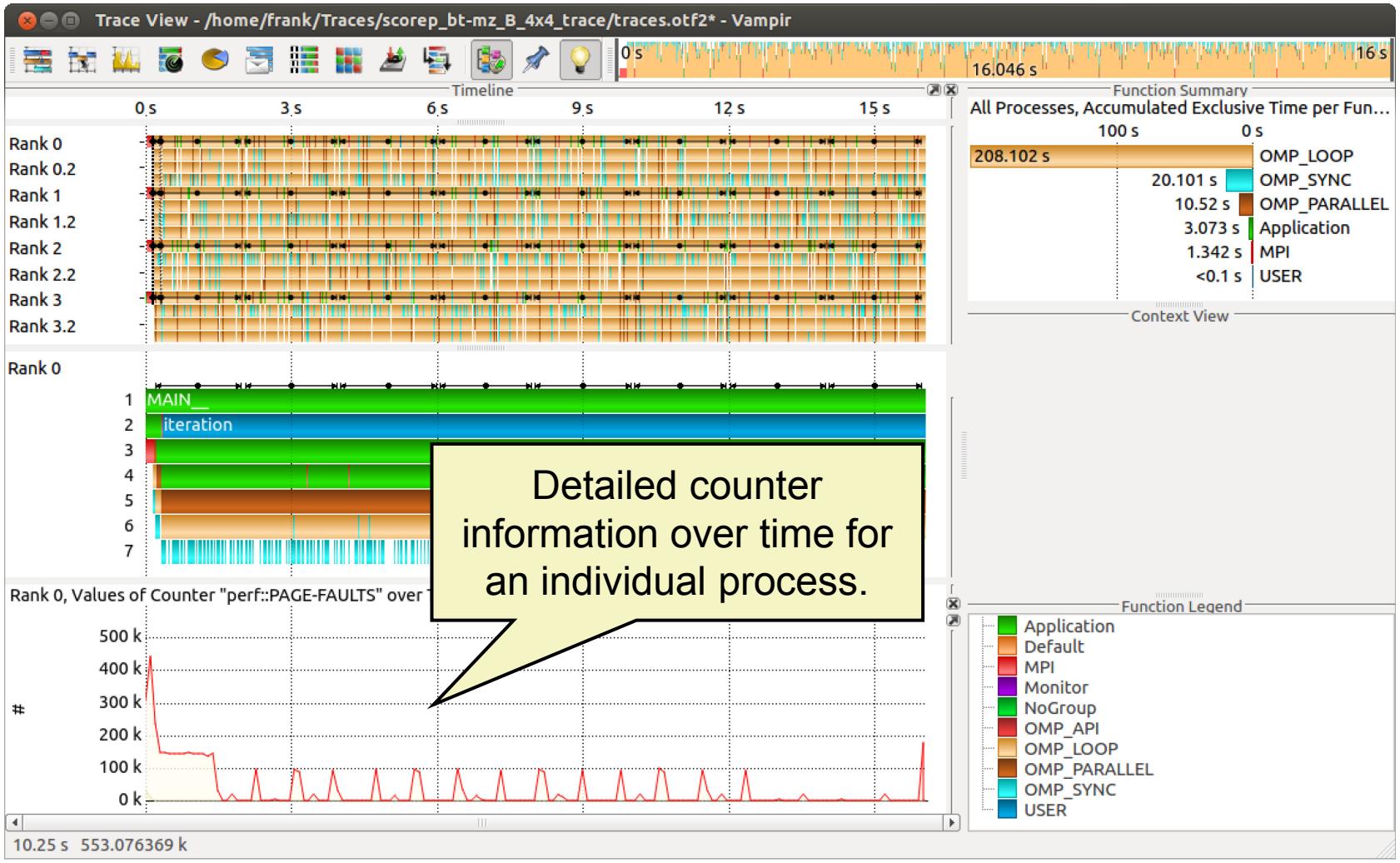
# Process Timeline



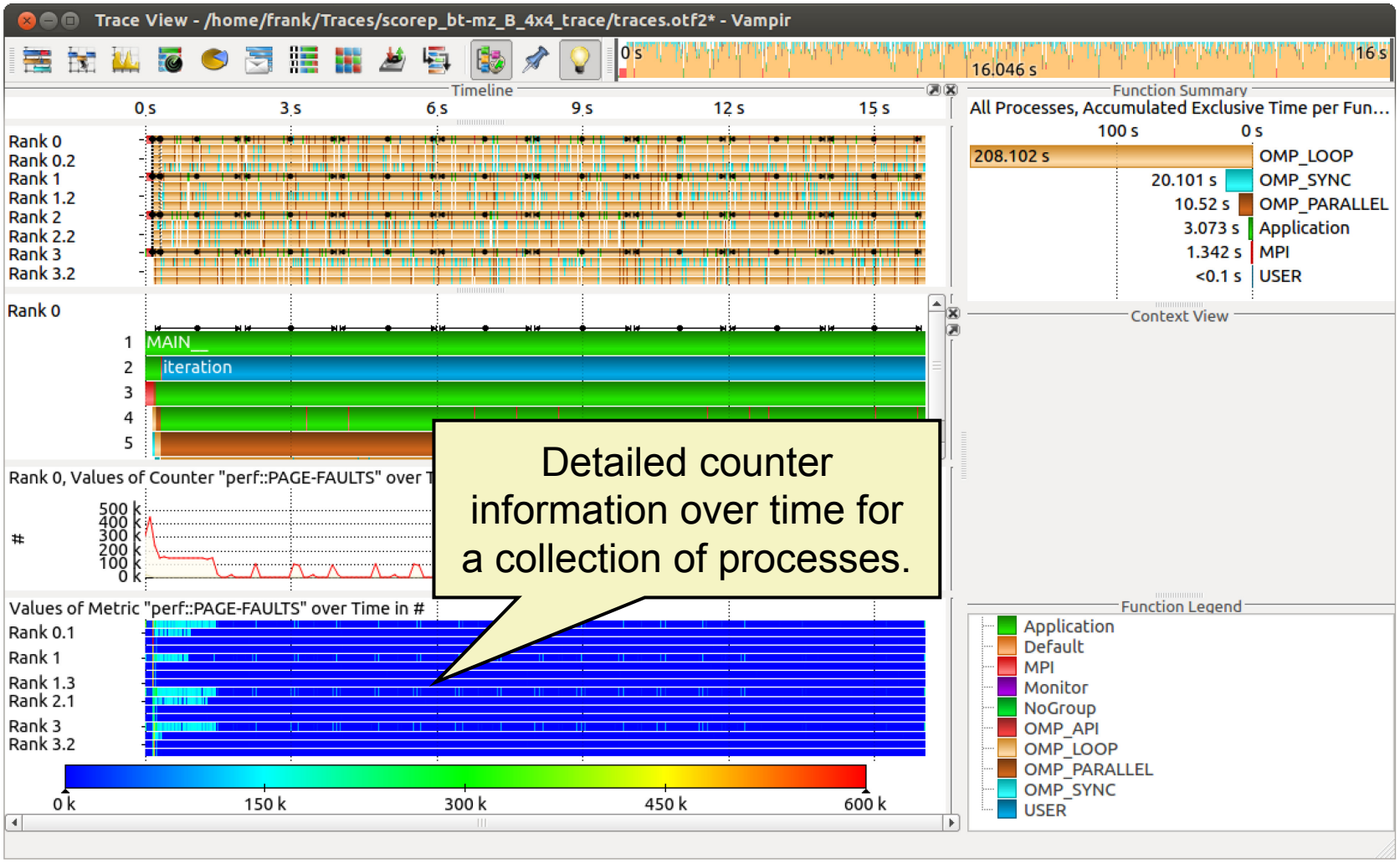
# Typical program phases



# Counter Data Timeline

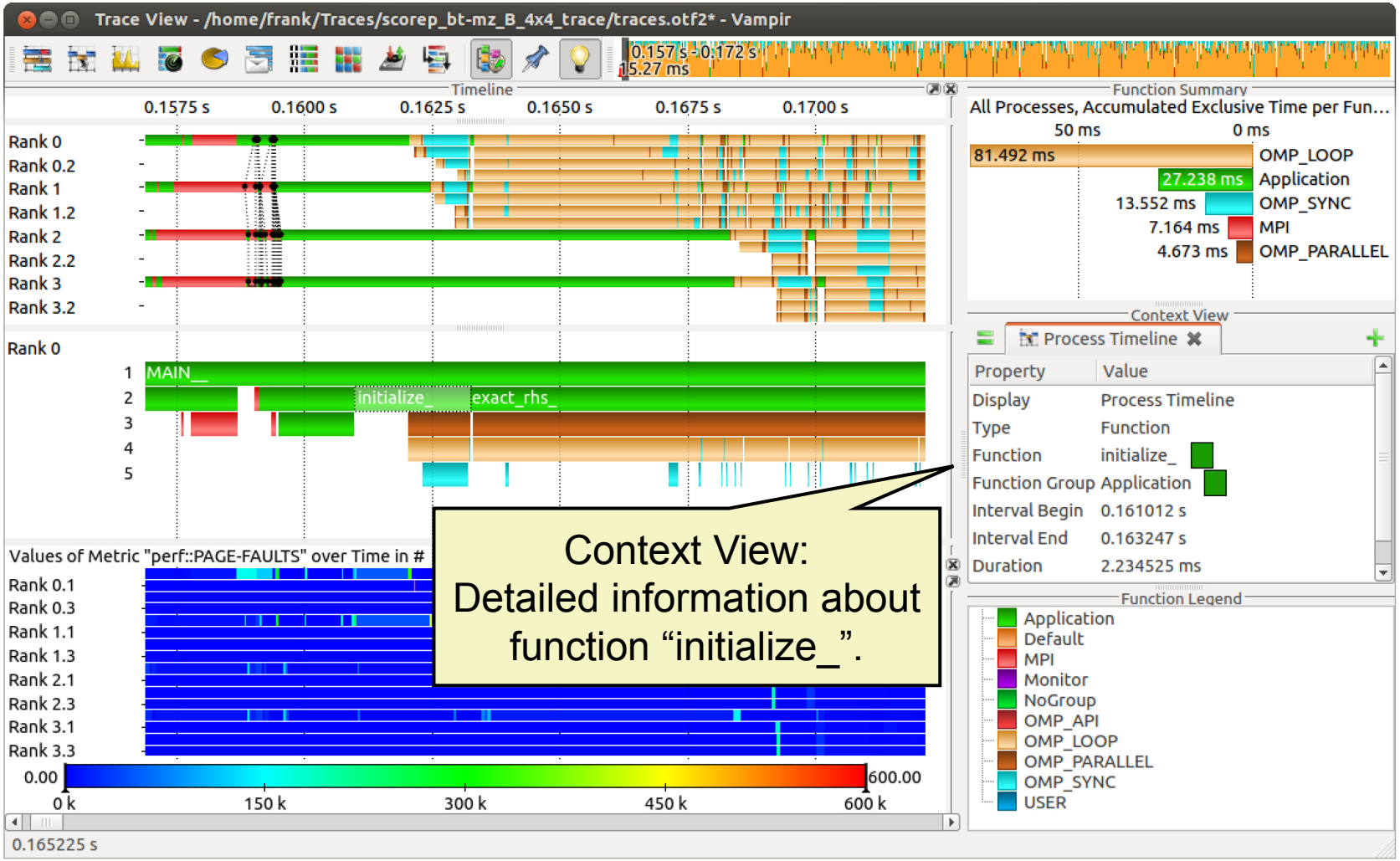


# Performance Radar

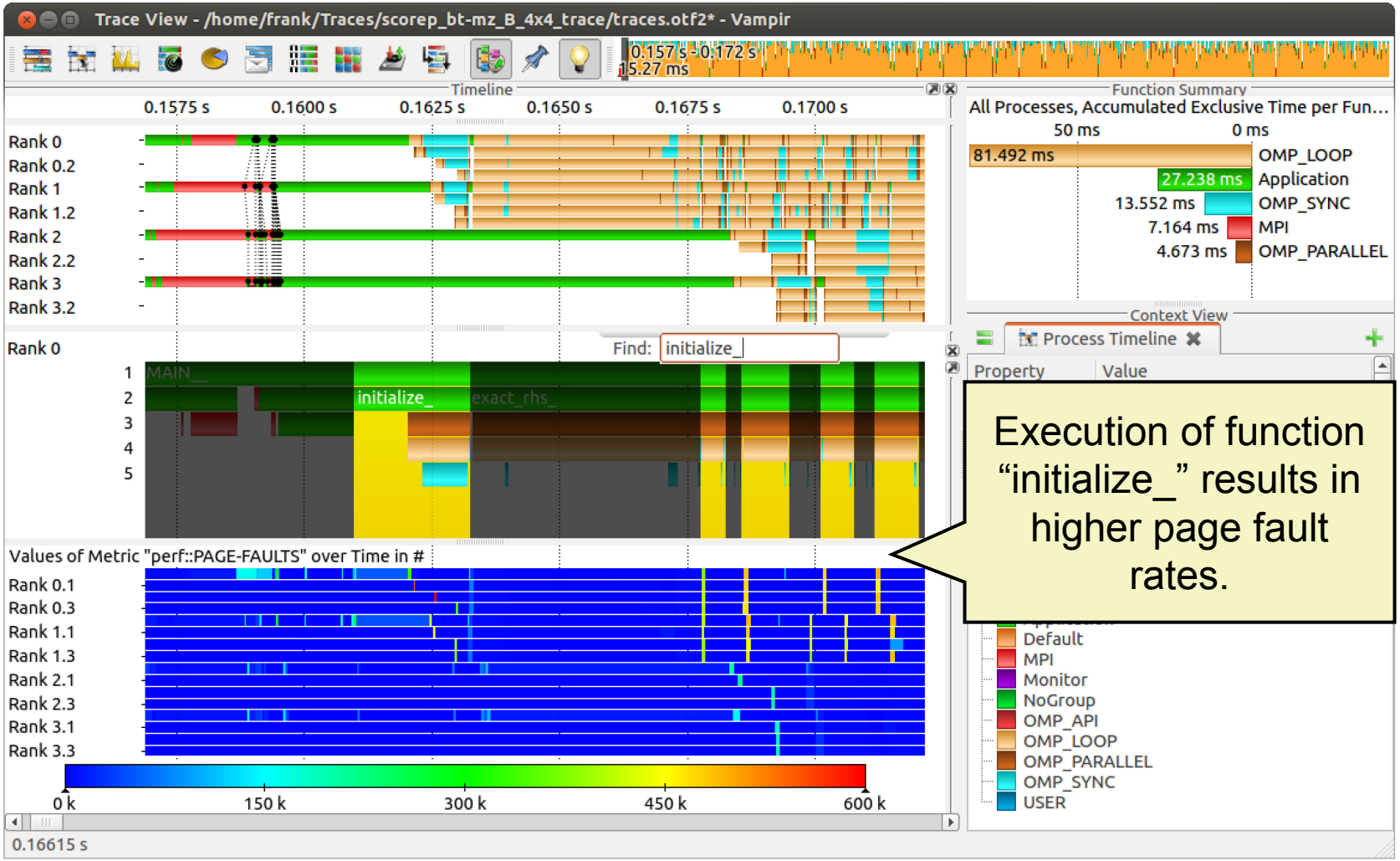


# Vampir: Visualization of the NPB-MZ-MPI / BT trace

## Zoom in: Inititialisation Phase



# Feature: Find Function

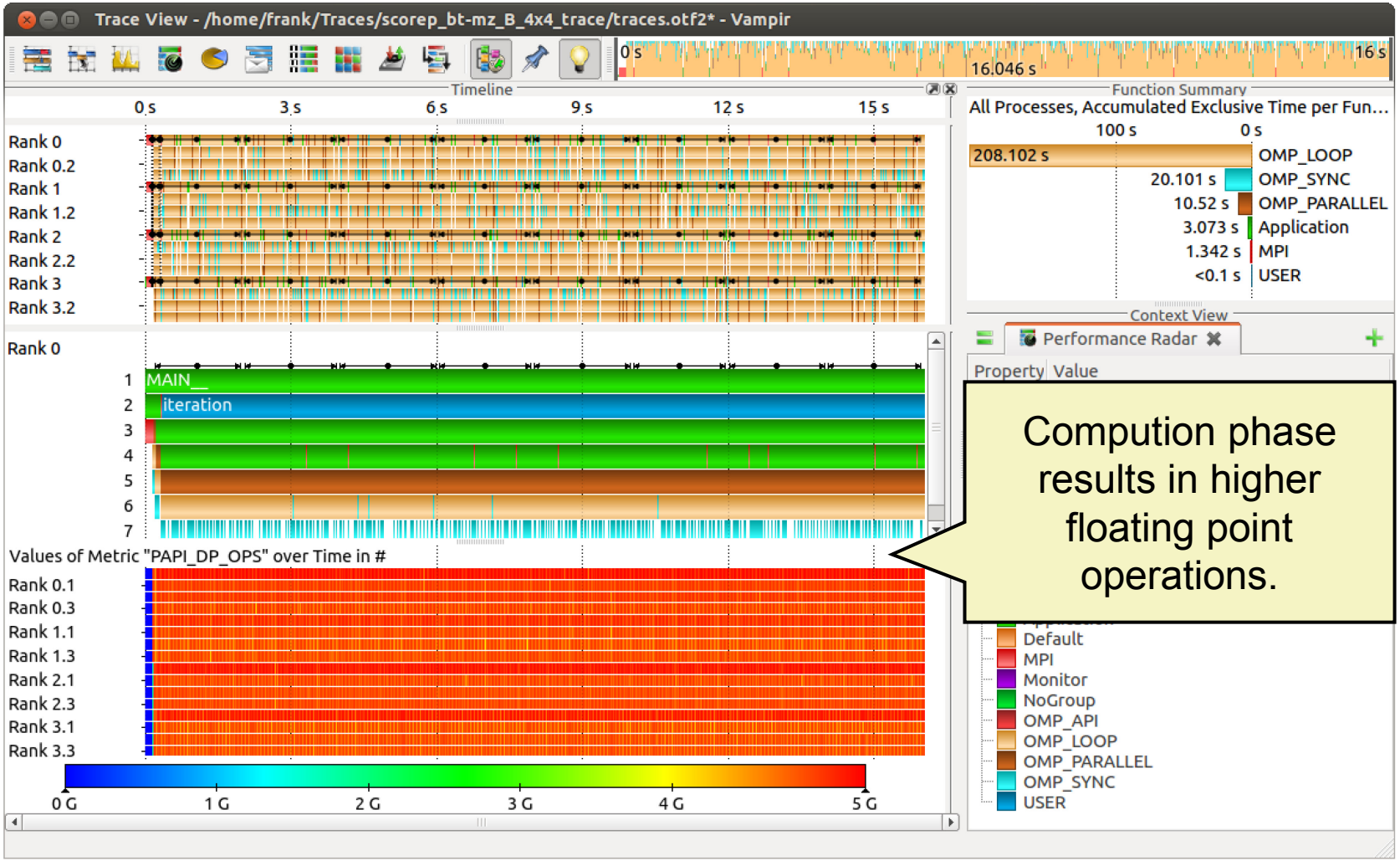


# Vampir: NPB-MZ-MPI / BT source snippet

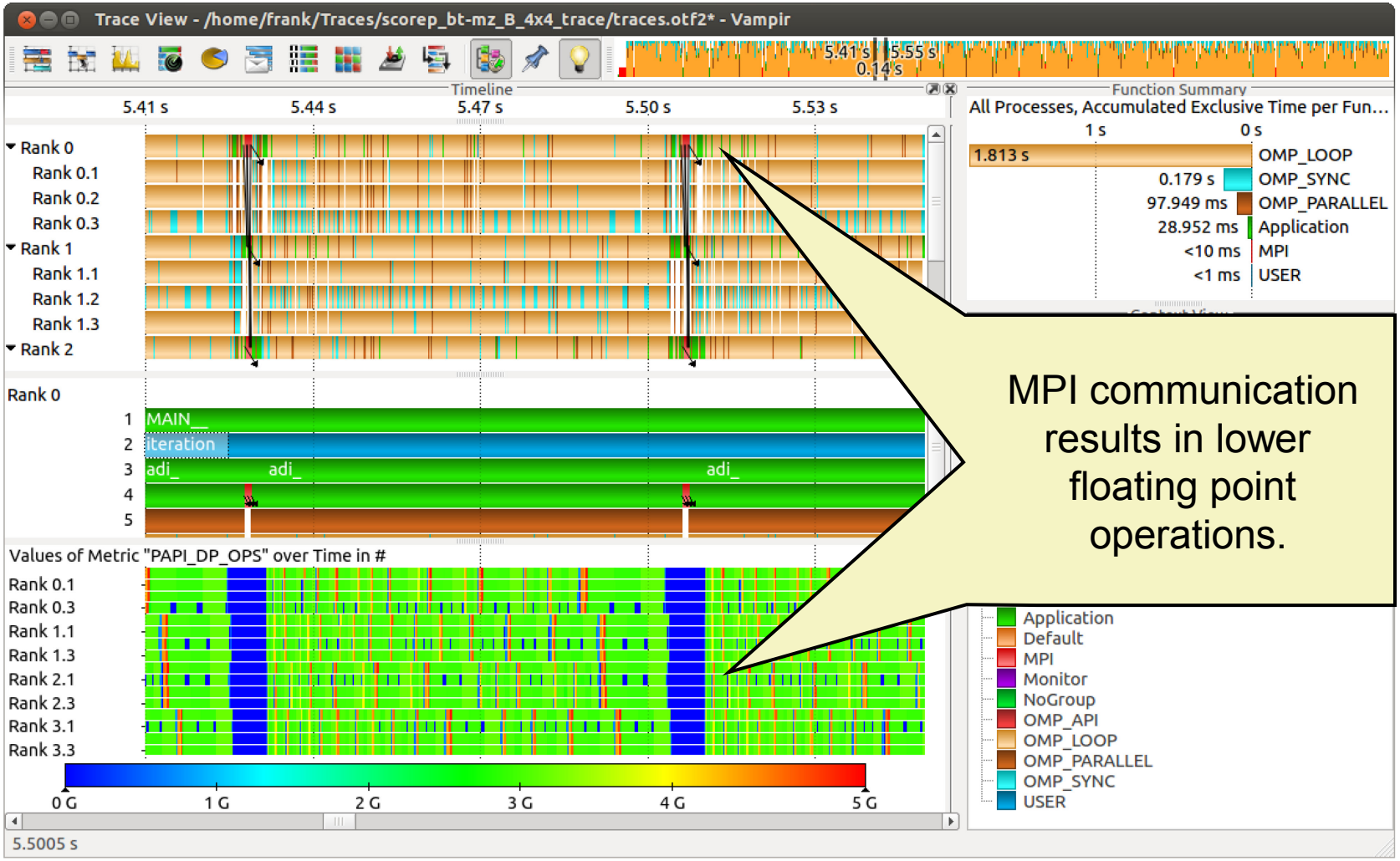
```
#include "scorep/SCOREP_User.h"
c---------------------------------------------------------------------
c      initialize data
c---------------------------------------------------------------------
       call timer_start(1)
       ...
c---------------------------------------------------------------------
c      start the benchmark time step loop
c---------------------------------------------------------------------
       do  step = 1, niter
         SCOREP_USER_REGION_BEGIN( my_region_handle,
                                   "iteration",
                                   SCOREP_USER_REGION_TYPE_COMMON )
        ...
         call exch_qbc(...)
         do iz = 1, proc_num_zones
           call adi(...)
         end do
         SCOREP_USER_REGION_END( my_region_handle )
       end do
       call timer_stop(1)
c---------------------------------------------------------------------
c      perform verification and print results
c---------------------------------------------------------------------
        ...
```

# Vampir: Visualization of the NPB-MZ-MPI / BT trace

## Computation Phase

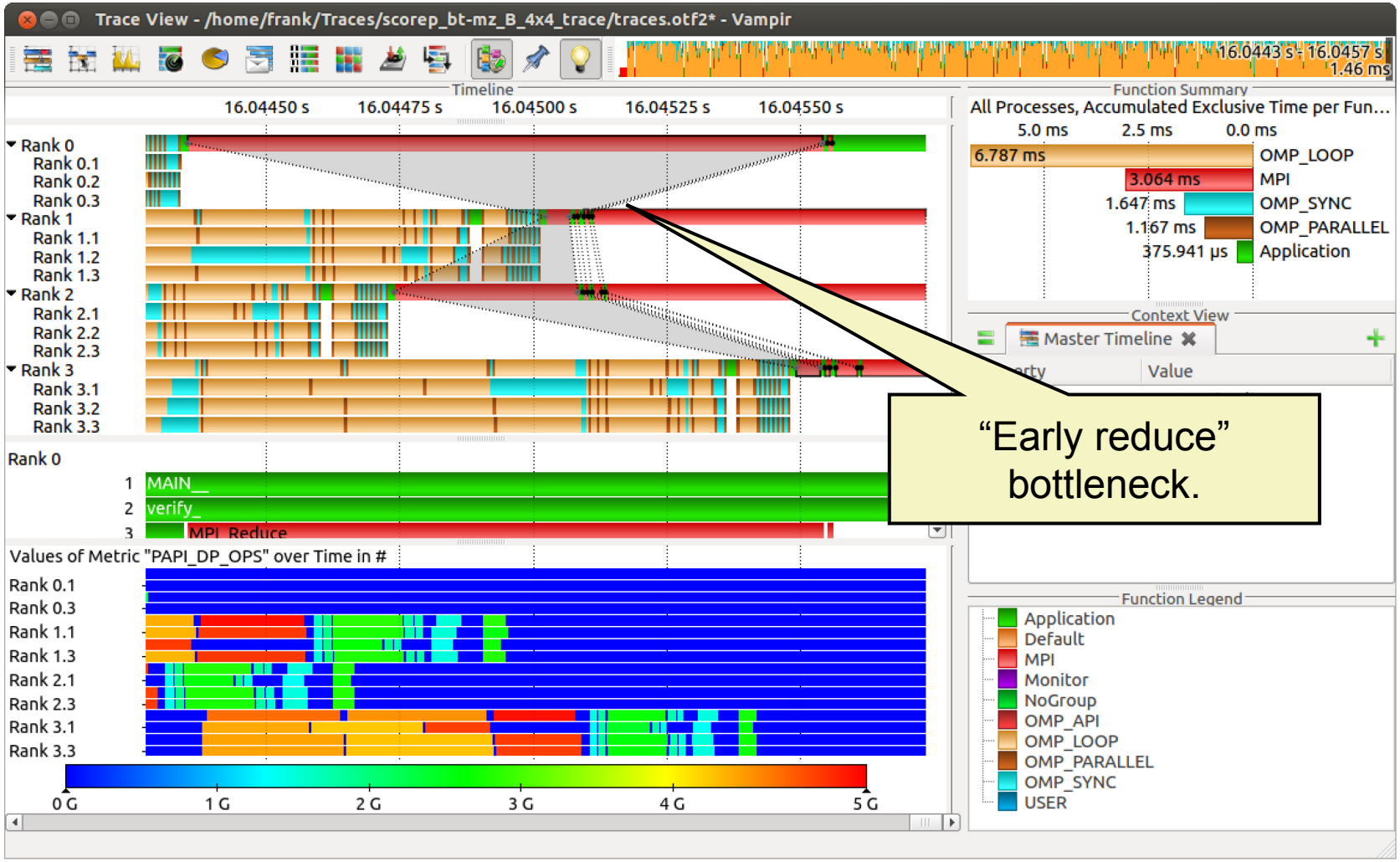


# Zoom in: Computation Phase



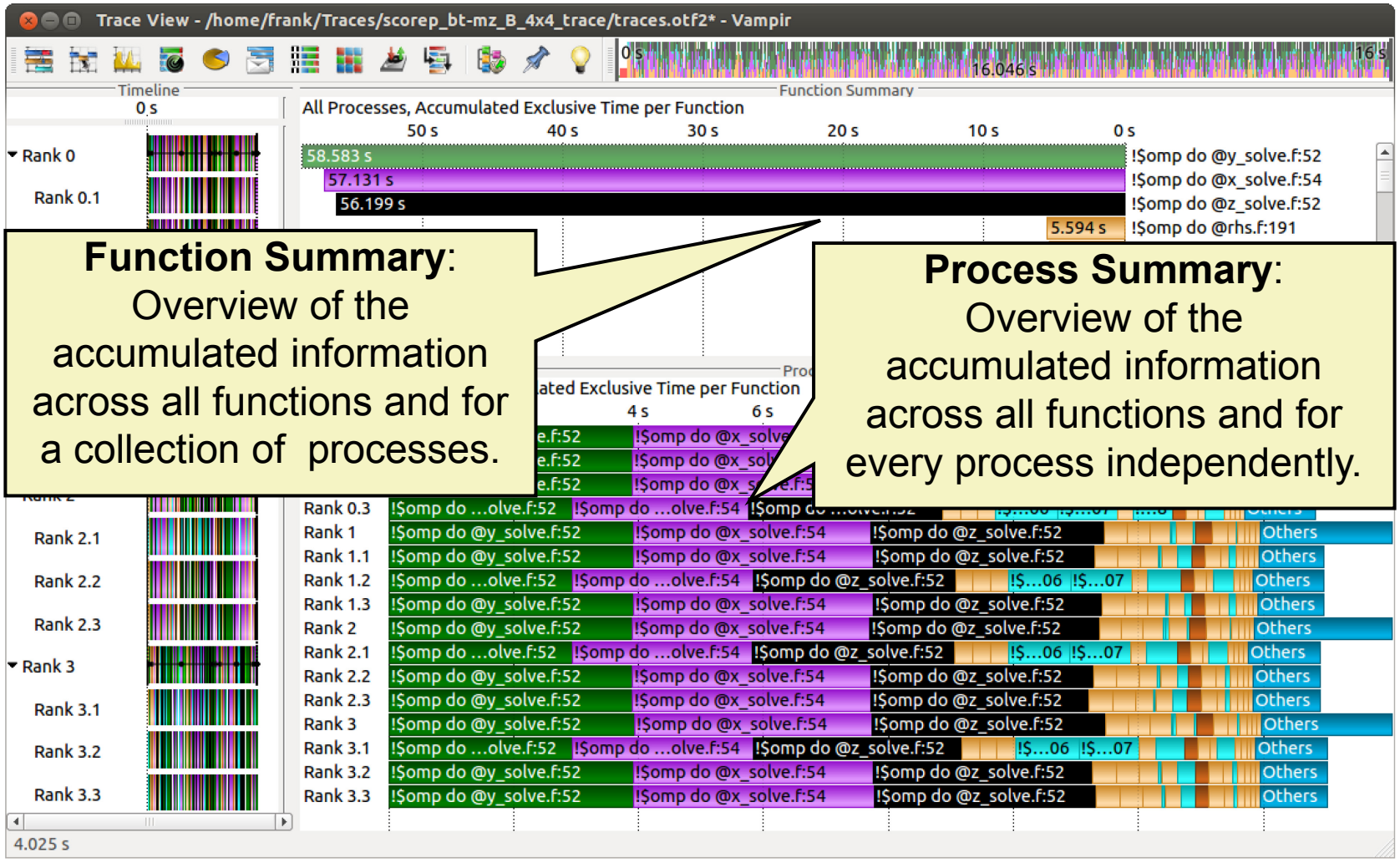
# Vampir: Visualization of the NPB-MZ-MPI / BT trace

## Zoom in: Finalisation Phase



# Vampir: Visualization of the NPB-MZ-MPI / BT trace

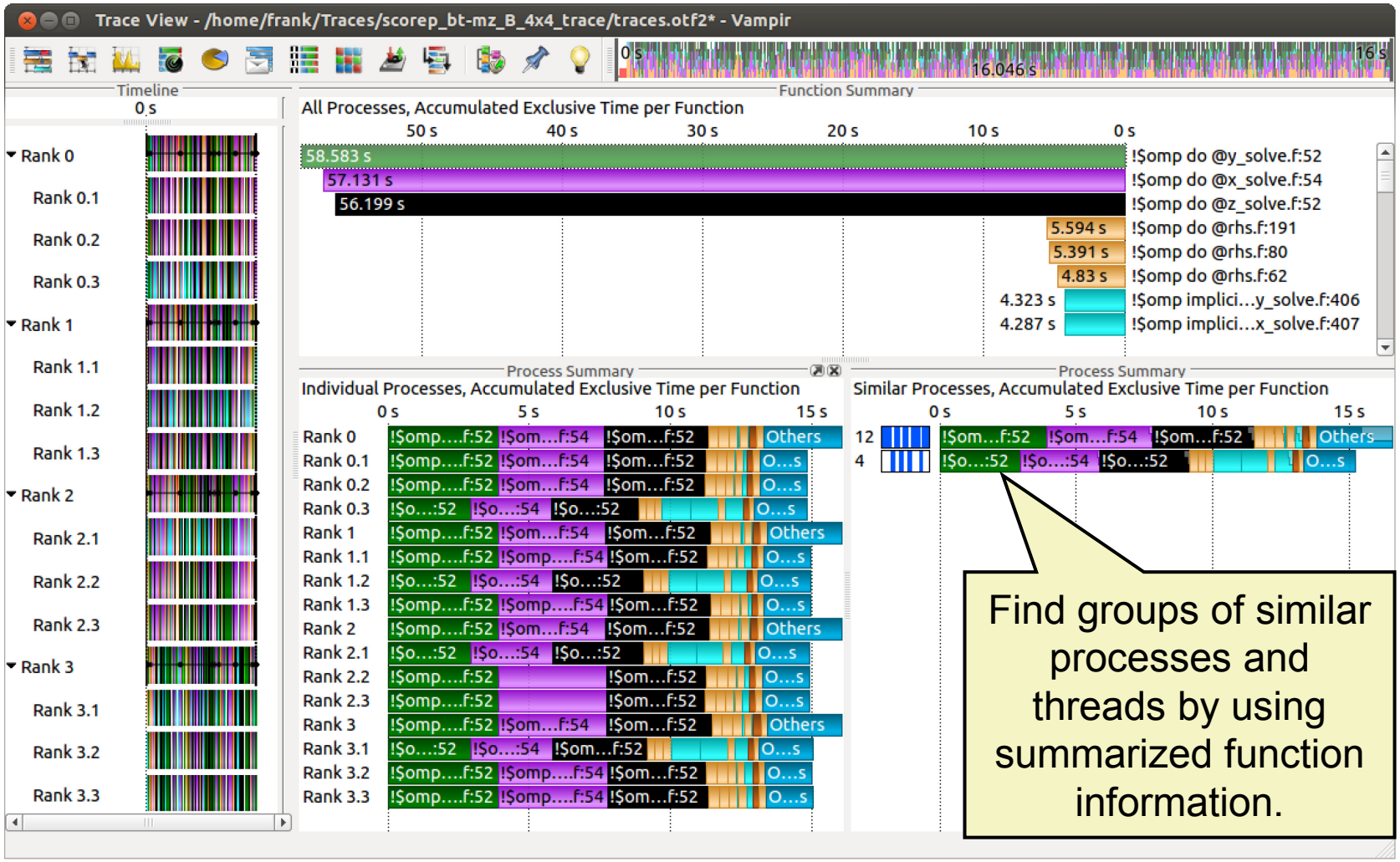
## Process Summary




**Function Summary**: Overview of the accumulated information across all functions and for a collection of processes.

**Process Summary**: Overview of the accumulated information across all functions and for every process independently.

# Process Summary




# Summary and Conclusion

# Summary

- Vampir & VampirServer
  - Interactive trace visualization and analysis
  - Intuitive browsing and zooming
  - Scalable to large trace data sizes (20 TByte)
  - Scalable to high parallelism (200000 processes)
- Vampir for Linux, Windows and Mac OS X

- **Note:** Vampir does neither solve your problems automatically nor point you directly at them. It does, however, give you FULL insight into the execution of your application.

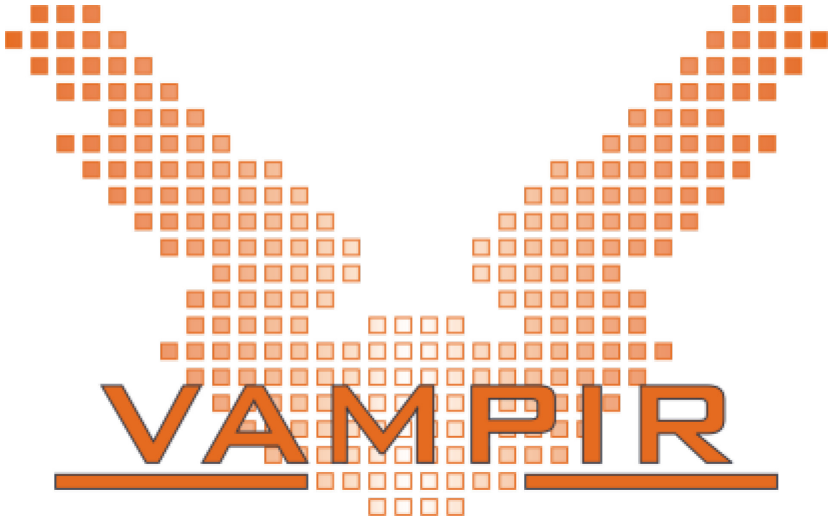
# Conclusion

- performance analysis very important in HPC

- use performance analysis tools for profiling and tracing
- do not spend effort in DIY solutions,
  e.g. like printf-debugging

- use tracing tools with some precautions
  - overhead
  - data volume

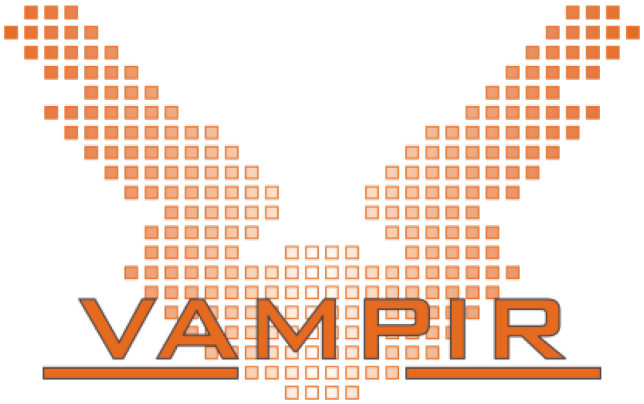
- let us know about problems and about feature wishes
- vampirsupport@zih.tu-dresden.de




Vampir is available at http://www.vampir.eu,

get support via vampirsupport@zih.tu-dresden.de

# Acknowledgement




Staff at ZIH - TU Dresden:

Ronny Brendel, Holger Brunst, Jens Doleschal, Ronald Geisler, Daniel Hackenberg, Michael Heyde, Matthias Jurenz, Michael Kluge, Andreas Knüpfer, Matthias Lieber, Holger Mickler, Hartmut Mix, Matthias Weber, Bert Wesarg, Frank Winkler, Matthias Müller, Wolfgang E. Nagel